Understanding the Context of Architecture Evaluation Methods

Len Bass
National ICT, Australia
Sydney, Australia
len.bass@nicta.com.au

Robert L. Nord
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pa, USA
rn@sei.cmu.edu

Abstract—This paper analyzes several industrial software architecture evaluation methods from the perspective of their contextual elements. Contextual elements include the time available for the evaluation, the personnel available, the openness of the results, the involvement of stakeholders, and so forth. The goal of this analysis is to inform managers and technical personnel about the possibilities for an architecture evaluation method given their particular context.

Keywords—software architecture evaluation methods, contextual factors, comparison of architecture evaluation methods

I. INTRODUCTION

Architecture evaluation methods have been documented to be in use since 1988 [16] and have been formalized since 1994 [12]. Since their introduction, multiple different evaluation methods have been proposed with some of them used in industry. Dobrica and Niemela provide a survey of methods up until 2000 [8]. They classified methods partially based on the goals of the evaluation. Some evaluation methods are specific for a particular quality attribute [5], some are specific for product lines [9], and some are specific for particular domains. Regardless of the goal of an evaluation, the context in which it takes place determines the feasibility of performing a particular evaluation on a particular system in a particular organization.

In this paper we explore the contextual factors of architectural evaluations. By contextual factors we mean items such as the time available for the evaluation, the participants in the evaluation, the involvement of stakeholders, the openness of the evaluation, and so forth. We restricted ourselves to methods that have had industrial and government use since the contextual factors are partially organizational and organizational influences tend not to be apparent in academic methods [11]. The goal of our analysis is to inform managers and technical personnel about the possibilities available to them given their particular context.

We begin by identifying the methods we are using as the basis of our analysis. Then we describe the measurement instrument that we are using to compare the methods, provide the contextual factors that we have identified and discuss them in terms of the various methods. We end by noting other comparison frameworks in the literature and discussing research questions involving architecture evaluation.

II. METHODS TO BE COMPARED

The industry methods that we will be comparing are the following: the Architecture Tradeoff Analysis Method® (ATAM®) [7], Tiny Architectural Review Approach (TARA) [19], Lightweight Architecture Alternative Assessment Method (LAAAM) [6], and Scenario-based peer reviews [3].

The ATAM is a method that helps a system’s stakeholders understand the consequences of architectural decisions with respect to the system’s quality attribute requirements. It is the evaluation method that TARA, LAAAM, and Scenario-based peer reviews use as a benchmark and has a special place in our methodology for defining a comparison framework.

TARA defined by Eoin Woods is based on industrial experience with the goal to provide guidance on how to run a simple architectural review to demonstrate the feasibility and the value of reviews in industry settings where scenario-based methods are unlikely to be successful.

LAAAM defined by Jeromy Carriere is based on tailoring the ATAM; it is scenario based and brings stakeholders together to define what quality means for the system and participate in a rational decision-making process.

Scenario-based peer reviews defined by Felix Bachmann is based on tailoring ATAM for iterative use during design. Instead of evaluating the entire architecture, a review finds risks associated with a single scenario.

III. CHOOSING A METHOD

From an engineering or a managerial perspective, choosing a method is a difficult proposition. Ideally, the academic community would structure experiments that take different types of systems, subject them to different evaluation methods, and compare the outputs and the costs of the various methods.

However, even structuring an experiment to compare two different applications of the same evaluation method is very difficult. You might think that evaluating the same system twice with different evaluation teams might provide some information but all of the methods except TARA rely on the architect for the system. Learning effects preclude the architect from objectively taking part in two evaluations. TARA is acknowledged to be subjective and depends heavily on the

* Architecture Tradeoff Analysis Method and ATAM are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.
expertise of the evaluator. Two different applications of TARA would have to use evaluators with equivalent expertise – a very difficult criterion to evaluate.

Relying on the author’s description of the methods to make a choice is equally problematic. Authors provide only anecdotal evidence about benefits. See the cited works for descriptions of the claimed benefits.

Authors will naturally express the benefits for their method and may not be aware of implicit contextual factors that inhibit or enable success. Absent scientific experiments and discounting author qualitative claims means that indirect techniques are needed in order to choose a particular method. We believe an investigation of context will provide some guidance to the manager and the engineer in the selection or creation of a method. That is the motivation for this paper.

IV. DEVELOPTMENT OF CONTEXTUAL FACTORS

In order to compare methods we need a conceptual framework for architecture review. We use the framework from the Software Architecture Review and Assessment (SARA) Report to provide a common vocabulary and model for understanding the context and workflow of the methods [18] since this framework currently represents the closest thing to a standard description of architectural evaluation.

SARA divides the context into four categories: inputs, outputs, mechanisms and controls. The inputs to a review are identified as architecture artifacts and other inputs and the outputs as the assessment report and other outcomes. Mechanisms are the review participants and the methods and techniques they employ. The controls are the review objects and policies and principles.

The evaluation activities of the process are divided into three phases: review inception, review, and post review. The inception phase results in the agreement on the review scope, cost, duration, participants, and so on. The review phase is the iterative process of discovering, capturing and comparing architecturally significant requirements and the architecture description. The post review phase summarizes and communicates the finding, as well as provides the opportunity for improving the review techniques and methods.

In order to compare methods using the SARA framework and isolate the effects of context, we use a prototypical method and isolate the assumptions within that method. We choose to use ATAM [7] for this purpose. This is not to necessarily advocate for ATAM. As many authors have pointed out, ATAM is heavy weight for their particular context. We choose ATAM as our prototypical method precisely because it has all of the steps needed to perform a thorough review. One of the points of this paper is that context will cause many to leave out steps from the ATAM or to cover them in a different fashion.

The assumptions of ATAM give us a candidate set of contextual factors. We then examine the methods to identify other factors that we might have missed. This process gives us our list of context factors – SARA provides the overall framework and vocabulary and the assumptions of ATAM as iterated through the other methods provide the contextual factors.

V. BASELINE/ATAM CONCEPTUAL FLOW

The steps of ATAM represent the evaluation process of the SARA framework. The inputs, outputs, mechanisms, and controls are referenced by the evaluation activities of the process.

1. Present the ATAM.
2. Present business drivers.
3. Present architecture.
4. Identify architectural approaches.
5. Generate quality attribute utility tree.
6. Analyze architectural approaches.
7. Brainstorm and prioritize scenarios.
8. Analyze architectural approaches.

The steps and their description can be found in [7].

From a process point of view, step 6 and step 8 of ATAM are identical. What distinguishes them is their context. The inputs to step 6 are scenarios from the utility tree and the design team participates in the analysis. The inputs to step 8 are scenarios from the brainstorm session and the entire group of stakeholders participates in the analysis. The selected methods generally conduct the analysis of architectural approaches in a single context rather than the two types of context done in ATAM.

Several assumptions are embedded in the ATAM steps.

1. Multiple stakeholders provide input to the evaluation process.
2. The requirements process may have been inadequate.
3. Documented analysis is based on comparing business goals to architecture using thought experiments.
4. The evaluation is conducted as an open process. Results are open as they are discovered. All attendees are of equal weight when prioritizing the scenarios.
5. The process is time limited.

VI. CONTEXTUAL ASPECTS

In this section, we identify the contextual aspects and the specific questions that we use to discuss our comparison of the methods. We frame the categories in terms of the questions one might ask to generate insight into a particular method and its deviation from the standard.

We derived the specific questions by turning the assumptions of ATAM into questions of the other methods. If the methods represent a weakening of ATAM, then some of the assumptions must be different. We then examined the questions against the other methods to determine whether anything had been missed. This caused us to add another factor to the SARA framework - process. When and how the process is executed is an important contextual element.
Our categories are:

- **Inputs.** An evaluation may have three kinds of inputs: a measure (or yardstick) for evaluation, a description of the target system to be evaluated, and a description of the functions of the system.

Some of the questions to be asked of these inputs are:

- Are the inputs explicit or implicit? Some requirements are explicit and others are not. Some systems have architectural documentation, others do not.
- When are they generated? Inputs can be generated as a portion of the evaluation or can be generated as a portion of the development process.

- **Outputs.** The output of an evaluation is a characterization of the system against the yardstick. Some of the questions to be asked are:

- How is the characterization framed? The characterization can be in terms of risks, effort to bring the system to a desired state and reach a go or no-go decision, a collection of recommendations for a path forward.
- Who sees the results of an evaluation? The results can be public for the participants or private for specific stakeholders.

- **Mechanisms.**

  - Who are participants?
  - What is the relationship of the evaluators in the evaluation to the organization – either the development team or a management organization? The evaluators may be inside the development team, inside the organization, or from outside the organization.
  - Who sets priorities (tradeoffs) for the system to be evaluated?
  - How do the steps of evaluation support or inhibit social interactions and affect the dynamics of the participants?
  - What are the resources required and what is the cost of the evaluation?
  - What skill level and knowledge are required by the roles of those participating?
  - What tools or automation are used during the evaluation?

- **Controls.**

  - What are the evaluation objectives? The objective can be the detection of problems, choosing among design alternatives, choosing among system alternatives.
  - What is the scope of the evaluation – the whole system or specific facets of the system?
  - **Process.**

- What is the schedule of the evaluation?
- How open is the process? Who is entitled to know what and when can they learn it?
- When is the evaluation held?

### VII. Contextual Analysis of Methods

Table 1 shows the methods and the answers to the specific contextual questions. From this table one can see the uniqueness of particular methods in their contextual assumptions.

- TARA is unique in several factors – requirements and architecture descriptions are generated as needed during the review, results are closed since they are sensitive and distribution is controlled by the sponsor, artifacts already exist, a static analyzer is used, one or more systems are the focus, and there is no face-to-face meetings among all stakeholders, although the evaluator may meet with individuals or small groups.  The evaluator is assumed to have domain expertise.

- LAAAM is unique in that it does not require an external evaluator although it does require an external facilitator. Part of the motivation for the method is that the outputs (quality tree and evaluation matrix) may live beyond the evaluation. As business priorities evolve, the quality tree can be revisited and the impact on the recommendation assessed. New stakeholders can absorb context from the evaluation matrix and bring their own perspectives as scenarios. As technology alternatives evolve, new choices can be added to the evaluation matrix.

- LAAAM and the Scenario-based peer reviews are intended to be part of the design process rather than an evaluation activity at the end of design. They share the characteristic of being a limited review. A portion of the design is evaluated rather than the whole design.

- Scenario-based peer reviews are unique in that the priorities are established prior to the review. Peer reviews are a continuous risk mitigation strategy to ensure the architecture design stays on track.

### VIII. Applying the Contextual Factors

The contextual factors give the manager and the engineer some basis for choosing a method. The following questions can assist in the choice.

- **When is the system to be evaluated?**
  - Design. LAAAM is designed to choose among design options. Scenario-based peer reviews are designed to be integrated with a design process.
  - Post design, pre implementation. ATAM is typically used after a design has been mostly completed but prior to implementation.
  - Post system construction. TARA is the only method that takes advantage of the existence of source code.
Table 1: Contextual factors in the various methods.

<table>
<thead>
<tr>
<th>Context factors</th>
<th>Specialized questions</th>
<th>ATAM</th>
<th>TARA</th>
<th>LAAAM</th>
<th>Scenario-based Peer Reviews</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inputs (architecture artifacts [target], other inputs [measure])</td>
<td>Explicit/implicit</td>
<td>Explicit business drivers, explicit description of the architecture</td>
<td>Implicit requirements, explicit description of the architecture</td>
<td>Explicit problem statement or scope definition, alternative solutions</td>
<td>Explicit description of the architecture necessary for the scenario under review</td>
</tr>
<tr>
<td>Generated prior to / during review</td>
<td>Prior to review</td>
<td>During review</td>
<td>Prior to review</td>
<td>Prior to review</td>
<td></td>
</tr>
<tr>
<td>Outputs (assessment report, other outcomes)</td>
<td>How is output framed</td>
<td>Risks, tradeoffs</td>
<td>How well architecture supports its key requirements, answers to sponsor questions</td>
<td>Choice of design options</td>
<td>Risks and to-do’s</td>
</tr>
<tr>
<td>Who sees output</td>
<td>Participants, commissioning manager</td>
<td>Commissioning manager</td>
<td>Participants, stakeholders up to executive management</td>
<td>Participants, project management</td>
<td></td>
</tr>
<tr>
<td>Mechanisms (review participants, methods and techniques)</td>
<td>Relation of evaluators to development organization</td>
<td>External</td>
<td>External to developers but internal to organization</td>
<td>External</td>
<td>External to development team</td>
</tr>
<tr>
<td>Priority setting</td>
<td>Within review</td>
<td>Within review</td>
<td>Within review</td>
<td>Prior to review</td>
<td></td>
</tr>
<tr>
<td>Social interaction</td>
<td>Face to face</td>
<td>None</td>
<td>Face to face</td>
<td>Face to face</td>
<td></td>
</tr>
<tr>
<td>Resources required</td>
<td>4 evaluators, architect, business rep, stakeholders</td>
<td>One person</td>
<td>One facilitator</td>
<td>Two evaluators</td>
<td></td>
</tr>
<tr>
<td>Skill level</td>
<td>Moderate</td>
<td>Expert</td>
<td>Moderate</td>
<td>Expert</td>
<td></td>
</tr>
<tr>
<td>Knowledge (of evaluators)</td>
<td>Architecture</td>
<td>Domain; architecture</td>
<td>n/a</td>
<td>Architecture</td>
<td></td>
</tr>
<tr>
<td>Tools or automation</td>
<td>Templates</td>
<td>Static analyzers</td>
<td>Specialized tool</td>
<td>Template</td>
<td></td>
</tr>
<tr>
<td>Controls (review objectives, policies and principles)</td>
<td>Evaluation objectives</td>
<td>Risk themes</td>
<td>Which system in a M&amp;A setting to continue</td>
<td>Design choice and rationale</td>
<td>Current state, risks</td>
</tr>
<tr>
<td>Scope</td>
<td>Whole system or major subsystem</td>
<td>List of competing systems</td>
<td>Specific design choices within system</td>
<td>One or two scenarios</td>
<td></td>
</tr>
<tr>
<td>Process (inception, review, post-review)</td>
<td>Schedule</td>
<td>~1 week prep time; One week total spread in 2 sessions over a month</td>
<td>5 days</td>
<td>3 90 minute sessions spread over a few to several weeks</td>
<td>2 hours every two weeks</td>
</tr>
<tr>
<td>Openness</td>
<td>Open to all attendees</td>
<td>Closed</td>
<td>Open to development team</td>
<td>Open</td>
<td></td>
</tr>
<tr>
<td>Within / after design</td>
<td>After</td>
<td>After (artifacts existed)</td>
<td>Before or within</td>
<td>Within</td>
<td></td>
</tr>
</tbody>
</table>

- **What level of expertise do you have available internally?** External evaluators are used for all the methods except LAAAM. Arranging for external evaluators takes time.

- **What explicit materials are available?** Only TARA allows for the generation of materials during the review. Preparing the materials takes time. Scenario-based peer reviews makes a virtue of the preparation by keeping the scope small and having multiple evaluations over different portions of the system. A template provides guidance to record results that are used to improve and show progress.

- **How involved do you want the stakeholders to be?** ATAM and LAAAM both make a virtue of stakeholder involvement by citing increased project communication.

The choice of a method may depend on factors other than the ones we have identified. Our contextual factors came from examining the various methods but organizational support for architecture evaluation may change which method is most appropriate.

TARA provides an example where the goal was to provide guidance on how to run a simple architectural review to demonstrate the feasibility and the value of reviews in industry settings. The reason for TARA's existence is to provide a first
step on the architectural evaluation journey for an organization that is skeptical about architectural evaluation. It is applicable where there isn’t organizational support and therefore the time and effort that can be expended on a review is limited. LAAAM has a similar motivation – educating the organization about the value of architecture reviews.

IX. CONCLUSIONS AND RESEARCH QUESTIONS

In this paper, we have derived a basis for comparison of architecture evaluation methods. The framework is based on abstracting from specific methods. This generalization technique for the generation of a framework is different than the specialization of more abstract comparison frameworks. Space precludes a discussion of other comparison frameworks but see [1, 2, 4, 13, 15, 17] for some others.

We began this research with the belief that there are fundamental techniques that every successful architecture evaluation method uses. Yet an investigation of the methods shows that this belief was unfounded. Explicit scenarios are a widely used technique for architecture evaluation but TARA does not use explicit scenarios. The evaluators most likely run through scenarios in their heads (hence the requirement for expert evaluators in those methods) but the methods themselves do not utilize explicit scenarios.

The data about the cost/benefit of architecture evaluations is sparse. One clear research question is the determination of cost/benefit data.

Another research question is the impact of corporate culture on the occurrence and results of an evaluation. Do the different corporate cultures affect how frequently and willingly projects undergo an evaluation? Some corporations are very reluctant to share negative information about their systems, even internally. Other corporations view publishing abstracted versions of results from a lack of the kind of information that enables an engineer to convince their manager to perform evaluations yet still not widely adopted.

In conclusion, architecture evaluation is an old technique yet still not widely adopted within industry. Partially this results from a lack of the kind of information that enables an engineer to convince their manager to perform evaluations or to convince the entire organization of the value of assessment. Understanding the contextual factors in the different evaluation methods provides some assistance to engineers in making their case to management.

X. ACKNOWLEDGEMENTS

We thank Felix Bachmann, Jeremy Carriere, and Eoin Woods for reviewing the draft results and providing valuable suggestions for improvement.

National ICT Australia is funded by the Australian Government’s Department of Communications, Information Technology, and the Arts and the Australian Research Council through Backing Australia’s Ability and the ICT Research Centre of Excellence programs.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

REFERENCES